自然言語処理 130－2
（1999. 3. 4）

日本語における漸進型書記素・音素アラインメント

スラヴェン・ビラチ、ティモシー・ボールドウィン、田中穂積

東京工業大学

情報理工学研究科

〒152－8552東京都目黒区大岡山 2－12－1
tel:81-3-5734-2831 fax:81-3-5734-2915

アブストラクト

本研究では、漢字を含んだ文字列を形態音素ユニット（「morpho-phonemic unit」）に分割し、それぞれのユニットを対応する読みがなにアラインする。これにより分割された書記素（「grapheme」）の読みを漸進的に学習し、その後の解析に応用する。本手法を小規模の正解セットによって評価した結果、98.29％の正解率が得られた。

Incremental Japanese grapheme-phoneme alignment

Slaven Bilac, Timothy Baldwin, Hozumi Tanaka

Tokyo Institute of Technology

Department of Computer Science

2-12-1,Ookayama,Meguro-ku, Tokyo, 152-8552, Japan
tel:81-3-5734-2831 fax:81-3-5734-2915
email:{sbilac,tim,tanaka}@cs.titech.ac.jp

Abstract

Given a kanji compound and associated kana-based reading, the proposed system incrementally segments the kanji compound into morpho-phonemic kanji units, and aligns each such unit with its corresponding reading. At each step of segmentation the grapheme-phoneme assignments are remembered for use in subsequent steps, thus forming an array of grapheme keys with their phonemic attributes. The system was evaluated on a limited manually segmented sample with 98.29% accuracy.

1. Introduction

This paper describes a system for incremental segmentation and alignment of Japanese graphemes with their corresponding phonemic units. Grapheme-phoneme ("G-P") alignment is defined as the task of maximally segmenting a grapheme compound into morpho-phonemic units, and aligning each unit to the corresponding substring in the phoneme compound (cf. "grapheme-phoneme translation" – Divay and Vitale, 1997; Bernstein and Nesly 1981; Vitale, 1991). The proposed system is first bootstrapped with the phoneme equivalents of individual grapheme characters, and then using this information it tries to determine the alignment scheme of grapheme strings more then one character in length. The process consists of several parse algorithms that incrementally segment the grapheme strings and store the phonemic correspondence information for use in subsequent parses, until the complete set of grapheme strings has been aligned.

For the purposes of this paper, 'grapheme string' refers to the kanji representation of a given word or a compound, and 'phoneme string' refers to the kana (hiragana and/or katakana) mora correlates. Although kana units (making up the Japanese syllabry) actually consist of one or more consonants followed by a vowel phoneme as well as the stand alone vowels ([a],[i],[u],[e] and [o]) and a single stand alone consonant ([n]), we have opted for using kana as our phonemic units in order to avoid consideration of phoneme combinatorial restrictions. It is important to note that despite being basically phonemic in nature, kana can also constitute a necessary part of grapheme strings. That is, kanji by themselves are not sufficient to represent Japanese language in full, as for example, conjugating suffices can only be represented by kana. As a consequence 'grapheme' representation includes both kanji and kana where appropriate, while 'phoneme' representation is always limited to kana characters.

'Maximal' segmentation simply means that aligned sections are segmented until they cannot be further subdivided into smaller meaningful units. Graphemes are segmented in such a way that each segment is a self-contained morpho-phonemic unit. In other words, conjugating parts of speech (verbs and adjectives) will be segmented so that the conjugating suffix is contained in the same unit as the stem. Such partitioning helps avoid discrepancies resulting from differing grapheme representation of the same basic string as in the case of *wa∘ri∘bi∘ki*[1] "discount" which can be presented by four grapheme variations (割引, 割*ri*引 *ki*, 割引 *ki*, 割*ri*引), hypothetically resulting in four phonemic assignments.

Furthermore, certain complete word strings cannot be presented in kanji resulting in basically identical kana representation on both sides of the G-P alignment, as is the case with postpositions, phonomimes, phenomimes and interrogatives, leaving no room for grapheme segmentation. As a consequence entries that do not contain full or partial kanji representation are automatically filtered out before the commencement of the segmentation process.

Machine-readable dictionaries (`MRD`) of Japanese provide the G-P tuples needed as

[1] To make this paper more accessible to readers not familiar with Japanese script, kana characters are written in Latin script throughout this paper, with character boundaries indicated by "∘" and segment boundaries indicated by "+".

input to the system, and in this research, the Japanese component of the EDICT dictionary[2] supplemented by entries and partial segmentation data from the Sinmeikai dictionary (Sinmeikai, 1981) was used. [3] To increase the initial assignment accuracy of the system, a pre-formatted unit kanji list annotated with regular phonological assignments was used as a seed.[4]

By generating a complete set of G-P alignments for dictionary entries we are able to assign a complete list of phonemic attributes of a given kanji, opening the possibility for major applications of our research in training a dynamic kanji tester as an aid to learners of Japanese. Another application would be in the examination of phonological changes, depending on combinations of kanji in a given grapheme compound(Ito and Mester, 1995; Tsujimura, 1996).

The remainder of this paper is structured as follows. Section 2 describes the process of incrementally generating the segmentation of G-P tuples, while the Section 3 provides evaluation of the system.

2. Grapheme-phoneme alignment

After reading in the initial list of kanji and their phonemic attributes the system also reads in a list of verbs with their G-P alignment(Ikehara *et al.*, 1997). At a later stage of processing grapheme segmentation candidates are checked against the entries in the verb list so as to determine plausibility of morpho-phonemic unit requirement for conjugated verb forms. At this point, the system preprocesses the dictionary entries to remove lexical ambiguity. The following step is to segment entries whose phonemic assignment can be trivially assigned. Finally, information gathered in previous steps is used to determine the best alignment for entries with irregular phoneme assignment. [5]

One vital aspect of the system is its ability to remember previous assignments of G-P tuples, and to keep increasing the database so as to include multi-character grapheme keys with their phoneme alignment. Initially the G-P alignment database consists only of single-character graphemes (kanji) and their corresponding phoneme representation, but through each parse the database is fortified with irregular phonemic attributes of the existing grapheme keys as well as with new grapheme keys and their respective phonemic assignments. 'Database' refers to the collection of all grapheme segments (single or multi-character) with all their phonemic attributes.

By preserving all the variations of the grapheme strings the system is able to retain a larger set of possibly identical phoneme assignments of lexically varied graphemes, and thus handle a larger number of cases likely to appear in open text, without violating the principle

---

[2] EDICT English-Japanese Dictionary. URL: ftp://ftp.cc.monash.edu.au/pub/nihongo/

[3] In the Sinmeikai limited segmentation of the phonemic representation is provided through an optional single space.

[4] EDICT KANJIDIC. URL: ftp://ftp.cc.monash.edu.au/pub/nihongo/

[5] For the purposes of this paper irregular phonemic attributes of a kanji character are all phonetically modified readings or special readings that appear only in combination with

of maximally segmenting the phonemes into morpho-phonemic units. In other words, given phoneme segments will always correspond to the semantic content of the grapheme aligned with them.

2.1 Pre-processing

Lexical ambiguity refers to the existence of multiple lexical "spellings" for a given phonetic content, all sharing the same basic semantic and kanji component. This can arise as a result of the possibility to replace kanji with their corresponding kana, or variations in *okurigana* (kana) suffices which can be conflated with or pulled apart from the stem kanji phonetic content. An example of this latter process can be seen for the verb *agaru* "to lift, raise" lexicalisable as either 上・*ru* or 上・*ga* ・*ru* , with the underlined *ga* kana character conflating with the kanji stem 上 in the former case for the same phonetic and semantic content. It is important to note that alternations never occur as prefixes of kanji. Also, there are exceptions (mostly in the names of places) where certain kana characters (usually '*no*' and '*ga*') are omitted from the grapheme component of the tuple while remaining in the phoneme part, even though they cannot be conflated with the kanji stem. For example, *ya*・*ma*+*no*+*te* can be lexicalized either by 山+手 or 山+*no*+手 even though underlined kana character *no* does not form a single morpho-phonemic unit with the kanji 山.

Given that the aim of this research is alignment of a grapheme (kanji) string with its phoneme content, we will ignore the effects of the possibility of replacing kanji with their corresponding kana in the grapheme string and take only *okurigana*-based lexical alternation into consideration during the analysis.

Any grapheme forms sharing the same phonetic content and only differing in *okurigana* alternates are first detected by comparing the grapheme and phoneme parts of the tuple. All alternations are then clustered together to form a single entry which will have a unique phonemic segmentation aligning with several variations of the grapheme representation. On completion of the segmentation assignment all the variations are equivalently aligned. This is achieved by finding the alignment for one of the extreme cases (either with maximal or minimal kana content in the grapheme string) and then extending the segmentation to remaining cases. The exception to this rule are phoneme segments realized through a process other than strict lexical alternation (see above). Alignment of such strings results in the possibility of having more segments in the phoneme than in the grapheme string; in case of *ya*・*ma*+*no*+*te* the alignment will read 山+手 rather than 山++手[6] since one segment has been completely removed from this variation of the grapheme string.

2.2 Segmentation constraints

The following phonological constraints on the segmentation procedure are used to

---

certain other characters and are not listed as proper readings of a character.

[6] Note that we require some mechanism for such cases to determine which grapheme segment delimiters align with which phoneme delimiters. While such indices are not indicated in alignment examples in this paper, they should be understood as being implicitly described for sequentially corresponding delimiters.

reduce the number of illegal alignments[7]:

> <p1> A demarkation in script form indicates a segment boundary, except for the case of kanji-hiragana boundaries [G]
> <p2> Kana in the grapheme string must align with direct kana equivalent in the phoneme string [G-P]
> <p3> The length of a kanji substring must be equal to or less than the syllable length of the aligned phoneme string [G,P]

Constraint *p1* requires that a segment boundary must exist at points where the script changes from kana to kanji and between hiragana and katakana. The exception is changeovers from kanji to hiragana that may or may not be distinguished as segment boundaries. This exception is necessary since conjugating parts-of-speech in Japanese (verbs and adjectives) usually consist of a single or multiple kanji character stem and a conjugating kana suffix. Therefore, in the case of verbs and adjectives only, kanji together with kana suffix can be considered a valid morpho-phonemic unit. In lexical alternation cases, by separating the suffix from the stem, we would get different segmentations depending on the grapheme realization. To come back to our previous example of *agaru* two different G-P alignment would be assigned for. <上+*ga*・*ru*> - <*a*+*ga*・*ru*> and <上+*ru*>-<*a*・*ga*+*ru*>.

Constraint *p2* makes sure that we are not violating the G-P correspondence of kana by aligning segments with different phonetic content. In other words, any kana appearing in the grapheme string must be aligned with its equivalent in the phoneme string. For example, in the case of *abunai "dangerous"* alignments such as <危+*i*> - <*a*・*bu* +*na*・*i*> are illegal.

Finally, *p3* requires that any character appearing in the grapheme string is aligned with at least one character in the phoneme string, that is, no character can be aligned with a blank string. This requirement prevents invalid assignments in cases like *kaze* "common cold" <風邪>-<*ka*・*ze*> where <風+邪>-<*ka*・*ze*+> would be a potential assignment since one of the regular phonemic attributes of 風 kanji character is given by the complete phoneme string. As stated above, the reverse does not hold as a character can appear in the phonemic realization of the string without having proper representation in the grapheme string.

2.3 Parsing and segmentation techniques

After the corpus has been pre-filtered to remove all grapheme strings that do not contain any kanji characters, an initial parse removes all instances of grapheme strings containing a single grapheme (kanji) and no kana. Since there is no room for lexical alteration or ambiguous alignment, the analysis task is trivial for these cases.

As a second step, graphemes containing a single kanji character and kana prefix and/or suffix are parsed. Constraints *p1* and *p2* help reduce the number of possible segmentations,

---

[7] [G] applicable to grapheme segmentation
[P] applicable to phoneme segmentation
[G-P] applicable to grapheme-phoneme alignment

and the final decision as to proper alignment is made after examining the single segment delimiter accessed from the Sinmeikai dictionary.[8] Thus, *damarikokuru* "fall silent" is parsed as <黙・*ri*+*ko*・*ku*・*ru*> rather than <黙・*ri*・*ko*・*ku*・*ru*> since the phoneme representation of the this entry in the Sinmeikai contains the segmentation marker.

Next, entries containing two kanji characters and no kana in their grapheme representation are targeted. At this stage, the system accepts the segmentation provided in the Sinmeikai for the phoneme component of the tuple as correct. Still, in cases where no segmentation is provided, the grapheme string is checked against the existing database contents for possible alignment information before being labeled as a unit entry in the database. As a result *hannou* "reaction" is segmented as <反+応> - <*ha*・*n*+*no*・*u*> and not <反応> - <*ha*・*n*・*no*・*u*> even though it could be argued that *nou* reading of 応 is a phonological variation of the regular reading *ou* interacting with *han*, and thus the full string constitutes a single morpho-phonemic unit. This decision was made to preserve consistency of keeping the record of phonological alternations at the smallest unit level possible. On the other hand in some entries, as in our *kaze* example above, the new reading is not a result of phonological change as it cannot be accounted for from unit kanji readings, but a full string-level phoneme assignment needs to be recorded on the compound level.

All remaining entries are then run through three recursive parsing algorithms responsible for assigning alignments in the following manner:

The first algorithm looks for maximal length segments of the grapheme possibly considerable as a morpho-phonemic unit. Segments are compared with the contents of the database while simultaneously being checked as conjugated forms of verbs or adjectives in order to evaluate their self-containedness, and recursively removed from the grapheme string until the whole string is parsed. For example, strings like *mitetoru* "realize" are successfully parsed as <見・*te*+取・*ru*> - <*mi*・*te*+*to*・*ru*> with this algorithm.

The second algorithm operates in a similar manner, with the only difference being that the initial input strings are maximal length lexical alterations as opposed to minimal ones used in the first algorithm. At this stage, segments whose minimal lexical alternations could not be recognized by the first algorithm, either because the conjugated form was not recognized or because the characters contained in the phoneme string could not be accounted for, are analysed. For example, <取・*ri*+立・*te*・*ru*> - <*to*・*ri*+*ta*・*te*・*ru*> could not be parsed by the first algorithm because the conflated *te* character (<取・*ri*+立・*ru*>) confused the system. However, the second algorithm successfully assigns the correct alignment.

Finally, the third algorithm checks possible segments for predictable phonological alternations resulting in phonemic assignments not previously listed in the database. In addition, instead of searching for maximal length segment this algorithm checks for any possible alignment structure that could satisfy the phonological constraints listed above without making sure that a different alignment could result in a smaller number of morpho-phonemically complete segments.

---

[8] Our system accepts the spacing suggested in Sinmeikai in this stage of the segmentation process, but ignores it at the later stages of the recursive parse algorithm. This discrepancy is mandated by the fact that Sinmeikai never provides more than one segment marker per entry, even though our system assigns up to six segments per entry.

3. Evaluation

The proposed system was tested on all G-P tuples from EDICT dictionary which were also listed in the Sinmeikai dictionary and contained at least one kanji character. Out of a total of 60,226 instances, the system was able to align 60,217 entries, with 9 unalignable entries leftover. An outline of the dictionary used in analysis is given below:

Total number of distinct entries: 60,226
Average grapheme string length: 2.35
Average kanji component of grapheme string: 1.93
Average kana component of grapheme string: 0.42
Average phoneme string character length: 3.98
Average phoneme string syllable length: 3.46

To our knowledge, there does not exist a solution set against which we could test the proposed system, thus forcing us to create a control set of our own. Due to restrictions on manpower, a complete annotation of all G-P tuples was deemed unfeasible. Therefore, a limited set of only 5000 tuples was randomly selected for manual annotation. Given the direct and indirect interaction between those 5000 tuples and the remainder of the dictionary, it is plausible to expect the behavior of the system in this restricted evaluation to be representative of the overall performance. The results of the evaluation are summarized in *Table 1*. Some entries (97 instances) could not be matched within the solution set, and are factored out of the evaluation.

From *Table 1*, we can see that the system performs exceptionally well in this limited test, with shorter grapheme strings (up to three characters) reaching accuracy of over 98.81% while the accuracy decreases with increasing length of the grapheme string. While this was expectable, it is important to note that if the system was tested against a dictionary containing a higher number of characters per grapheme string accuracy levels would probably be lower.

Most of the mistaken assignments are result of the inability to properly recognize conjugated verb forms (30 instances), thus yielding a higher number of segments than necessary. Inability to recognize further indivisible two kanji character compounds is responsible for another 17 instances, while the remainder is caused by a variety of factors.

The system was also tested without using the segmentation clues provided in Sinmeikai, and the alignment accuracy was 92.90%.

Grapheme string length	Annotated solution	Correct solutions	Incorrect solution	Accuracy (%)
1 character	401	401	0	100.00
2 characters	3170	3155	15	99.53
3 characters	898	860	38	95.77
4 characters	314	289	25	92.04
5 characters	109	106	3	97.25
6 characters	9	8	1	88.89
7 char. and up	2	0	2	0.00
Total	4903	4819	84	98.29

Table 1: Comparison of the G-P tuple segmentation assignments with manually annotated solutions

4. Conclusion

The aim of this paper has been to present a system that generates grapheme-phoneme alignments based on both bootstrapped reading seeds and incrementally learned information. While the system performs well in the limited test setting it remains to be determined how it will perform on larger dictionaries containing entries of higher character length.

Continued research is needed as several problems persist. The system is somewhat dependant on the segmentation provided within the augmented dictionary entries, as well as being unable to handle many cases of conjugated verb forms with lexical alternates. By removing all the dependency on partial annotation of the dictionary input, the system would be able to tackle harder problems like open text with much more success. Problems of deflated lexical alternates keep evolving as more liberal grapheme representations find their way into daily language.

Still, it is our hope that the system will prove itself useful in related research topics. For one, we feel that system can be used as a basis for work on kanji teaching and testing tools for students of Japanese language and also for further study on phonological changes of kanji attributes. We intend to make publicly available the complete list of grapheme keys with their phonemic attributes.

References

Jared Bernstein, Larry Nesly. Performance Comparison of Component Algorithms for the Phonemicization of Orthography. In *Proceedings of the 19th Annual Meeting of the Association of Computational Linguistics*, pp.19-22, 1981.

Michel Divay, Anthony J.Vitale. Algorithms for Grapheme-Phoneme Translation for English and French: Applications for Database Searches and Speech Synthesis. *Computational Linguistics*, Vol 20, No.4,pp.495-523, 1997.

S. Ikehara, M. Miyazaki, A. Yokoo, S. Shirai, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi. *Nihongo Goi Taikei –A Japanese Lexicon.* Iwanami Shoten. 1997. (In Japanese)

Junko Ito, Armin R. Mester. Japanese Phonology in J.A. Goldsmith (ed.) *The Handbook of Phonological Theory*, 1995.

Sinmeikai Dictionary. Sanseido Publishers, 1981.

Natsuko Tsujimura. *An Introduction to Japanese Linguistics* Blackwell Publishers, 1996.

Tony Vitale. An Algorithm for High Accuracy Name Pronunciation by Parametric Speech Synthesizer. *Computational Linguistics*, Vol 17, No.3,pp. 257-276, 1991.